Decoder Synchronization For Deep Space Missions

J. I. Statman, K.-M. Cheung, T. H. Chauvin, J. Rabkin, and M. L. Belongie
Communications Systems Research Section

The Consultative Committee for Space Data Standards (CCSDS) recommends that space communication links employ a concatenated, error-correcting, channel-coding system in which the inner code is a convolutional (7,1/2) code and the outer code is a (255,223) Reed–Solomon code. The traditional implementation is to perform the node synchronization for the Viterbi decoder and the frame synchronization for the Reed–Solomon decoder as separate, sequential operations. This article discusses a unified synchronization technique that is required for deep space missions that have data rates and signal-to-noise ratios (SNRs) that are extremely low. This technique combines frame synchronization in the bit and symbol domains and traditional accumulated-metric growth techniques to establish a joint frame and node synchronization. A variation on this technique is used for the Galileo spacecraft on its Jupiter-bound mission.

I. Introduction

The traditional approach to decoding the channel error-correcting coding in the space communication links [1] is for the implementation to follow the CCSDS functional model [2] shown in Fig. 1, i.e., establish a concatenated decoder consisting of two distinct stages: a Viterbi decoder and a Reed–Solomon decoder, with no feedback between the two stages. Each of the two decoders requires appropriate synchronization: the Viterbi decoder requires node synchronization (the grouping of n-tuplets of soft symbols that correspond to a single information bit), and the Reed–Solomon decoder requires frame synchronization (the detection of the transport frame and the extraction of Reed–Solomon words). In most applications, data received prior to the accomplishment of synchronization are lost; however, as long as the symbol-signal-to-noise ratio (SSNR,

E_S/N_0) is relatively high, the synchronization time is short and the data loss is often ignored.

For a deep space communications environment, such an approach is often deficient—E_S/N_0 could be low, hence the synchronization time, measured in number of bits, would be longer. As the data rate decreases, the synchronization time, fixed in terms of number of bits, can result in loss of a significant percent of total data. Also, the sequential nature of the synchronization process compounds the data loss. To speed the synchronization process and reduce the data loss, we introduce here a joint synchronization technique that is being applied in the ground support for the Galileo deep space mission to Jupiter [3].

Section II introduces the core algorithms and the joint synchronization approach. Section III discusses the ap-

plication of joint synchronization in several scenarios, and Section IV presents two specific cases where the joint synchronization approach is applied.

II. Description of Synchronization Algorithms

The structure of the general joint synchronization decoder is shown in Fig. 2. It consists of the traditional series of processing functions, namely a Viterbi decoder, a deinterleaver, and a Reed–Solomon decoder, preceded by a soft symbol buffer. The processing functions are controlled by a joint synchronization function, which in turn relies on several core synchronization algorithms. In this article, we select three such core synchronization algorithms: a frame-marker correlator in the symbol domain, a frame-marker correlator in the bit domain, and an accumulated-metric growth-rate indicator. It is worth noting that decoders often contain other synchronization indicators that can be integrated into a joint synchronizer using techniques similar to those described below.

A. Bit-Domain Correlation

This algorithm examines the Viterbi-decoded bits at the output of the Viterbi decoder for presence of the frame marker pattern. When the pattern is detected, it indicates that the node synchronization hypothesis is "true" and, by definition, frame synchronization was accomplished.

Let the transport frame be of length M, including a frame marker of length N at the beginning of each frame. Let the frame marker be $\langle d_o(i),\ i = 0,\ \dots,\ N-1,\ d_o(i) = \pm 1 \rangle$. Then, the transmitted data stream can be represented as

$$D(i) = \left\langle \begin{array}{ll} d_o(i \bmod M), & i \bmod M < N \\ data, & i \bmod M \geq N \end{array} \right\rangle \quad (1)$$

The bit-domain correlation algorithm computes the running correlation, $J_{bits}(k)$, between the frame marker and the received signal $s(i)$

$$J_{bits}(k) = \frac{1}{N} \sum_{i=0}^{N-1} d_o(i) s(i + k) \quad (2)$$

where $s(i)$ is $D(i)$ contaminated by the effects of transmission, reception, and decoding. The expected value of $J_{bits}(k)$ is 1 when the received sequence is perfectly aligned with the frame marker. Elsewhere, for a properly selected frame marker, its autocorrelation properties ensure that

the expected value of $J_{bits}(k)$ is near zero. In practice, there are two methods to implement the bit-domain correlation algorithm:

(1) Testing for threshold, where the value of $J_{bits}(k)$ is compared against a threshold for a large number of k's. When the value of $J_{bits}(k)$ exceeds the threshold for the first time at k_0, the hypothesis that $s(i + k_0)$ is the beginning of the frame marker is declared true.

(2) Testing for maximum, where the value $J_{bits}(k)$ is maximized over all $0 < k < M$, regardless of a threshold. Let k_0 be the location where $J_{bits}(k)$ reaches its highest value over the search range; then the hypothesis that $s(i + k_0)$ is the beginning of a frame marker is declared true.

In the special interest of *transparent* convolutional codes (i.e., if the soft symbols are inverted, the Viterbi-decoded bits are inverted as well), $J_{bits}(k)$ may take the values of $+1$ for "correct symbol phase correlation" or -1 for "inverted symbol phase correlation," indicating that the Viterbi decoder is synchronized but the soft symbols are inverted and must be returned to the correct phase (i.e., reinverted) prior to deinterleaving. For transparent codes, implementing the bit-domain correlation requires searching for both a maximum and a minimum of $J_{bits}(k)$, or alternatively employing high and low thresholds.

The bit-correlation approach is powerful and usually highly reliable. Its main disadvantage is that it requires hypothesis testing at all possible offsets of the soft symbol n-tuplets, as well as the correct and inverted symbol phases (only for nontransparent convolutional code). Thus, for a $(15,1/6)$ nontransparent convolutional code, a large number of attempts (12 attempts in the worst case, 6 attempts on the average) are required before synchronization is accomplished.

B. Symbol-Domain Correlation

This algorithm examines the soft symbols prior to the Viterbi decoder for presence of an encoded version of the frame marker pattern. When the pattern is detected, it provides both the node and frame synchronization *prior to any decoding operation*. In this case, the symbol correlation function is

$$J_{sym}(k) = \frac{1}{N - (K - 1)} \sum_{i=K}^{N} \frac{1}{n} F_e(d_0(i)) S(i + k) \quad (3)$$

where $F_e(d_0(i))$ is the n-tuplet of soft symbols corresponding to a single bit of the frame marker, $S(i + k)$ is an n-tuplet of received symbols, and the operation on the two

n-tuplets is a dot product. The main disadvantage of this algorithm is that correlation can be performed only over part of the frame marker. For a frame marker of N bits and convolutional code of length K, only the symbols corresponding to the last $N - (K - 1)$ bits are known in the symbol domain—the symbols that correspond to the first $K - 1$ bits of the frame marker are corrupted by the unknown previous contents of the encoder. Depending on N, K, and E_s/N_0, the partial correlation may degrade the correlation SNR sufficiently to make synchronization difficult.

To assess the performance of synchronization using symbol-domain correlation, let us compute the probability of false detection. Let the soft symbol be modeled as having a value of $\pm m + n_{sym(i)}$, where $\pm m$ has an equal probability of being $+m$ and $-m$ and $n_{sym(i)}$ is $N(0, \sigma_{sym})$.[1] Let us assume that the frame marker has been selected such that its autocorrelation is near ideal, i.e., no significant secondary correlation peaks exist. Let us further assume that the correlation between the frame marker and a noiseless set of received symbols contains no significant peaks. Then, when the received symbols are not aligned with the frame marker, $J_{sym}(k)$ can be modeled as $N(0, \sigma)$,

$$\sigma = \sigma_{sym}/\sqrt{n(N - (K - 1))} \qquad (4)$$

while when the received symbols are aligned with the frame marker, $J_{sym}(k)$ is modeled as $N(m, \sigma)$. The probability of selecting an incorrect peak is given by

$$P_{FD} = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-m)^2}{2\sigma^2}}$$

$$\times \left[1 - \left(\int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{2\sigma^2}} \, dy \right)^{M-1} \right] dx \qquad (5)$$

If the effect of correlation between a random symbol pattern and the frame marker is included, the resulting probability of selecting an incorrect peak is given by Eq. (6) below. Figure 3 plots P_{FD} as a function of the correlation SNR, $10\log (m^2/2\sigma^2)$

[1] A zero-mean, normally distributed random variable with a standard deviation of σ_{sym}.

$$P_{FD} = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} e^{-\frac{(x-m)^2}{2\sigma^2}}$$

$$\times \left[1 - \left(\frac{1}{2^N} \sum_{i=0}^{N} \binom{N}{i} \right. \right.$$

$$\times \left. \left. \int_{-\infty}^{x+N-2i} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{2\sigma^2}} \, dy \right)^{M-1} \right] dx \qquad (6)$$

C. Accumulated Metric Growth Rate

This algorithm examines the accumulated metric at all the Viterbi decoder states, indicating the level of "mismatch" between the received soft symbol stream and the bit stream associated with the specific state. Even though the accumulated metric varies from state to state, its peak-to-peak variation is bounded by

(constraint length $-$ 1)(max branch metric)

Because of this bound, implementors often monitor the growth rate of a single selected accumulated metric where at high Eb/N_0 there is a clear distinction between in-node-synchronization and out-of-node-synchronization conditions. The distinction between the two conditions becomes more blurry as Eb/N_0 decreases; the thresholds for detecting the in-lock and out-of-lock hypotheses must be chosen carefully to meet the probability-of-detection and false-alarm requirements. Unfortunately, threshold selection must be accomplished empirically, as the growth rate measurement does not lend itself to analytic expressions.

III. Unified Synchronization

The objective of the unified synchronization approach below is to realize the attractive benefits of the symbol-domain correlation; namely, accomplish node and frame synchronization prior to the Viterbi decoder, while mitigating the algorithm's degraded performance at low SNR.

Case 1: Independent frames, minimal restriction on latency. Let us first observe that many communications links consist of transport frames that are independently encoded as a result of the fact that the frame synchronization marker is longer than the constraint length of the convolutional code, therefore serving as an effective barrier between the frames, resetting the encoder to

a known state. Let us also assume that there are minimal restrictions on latency.[2] In this case, a simple decoder architecture is possible,[3] as shown in Fig. 4.

In this architecture, no decoding is initiated until the input stream has been separated into frames, using frame-marker correlation in the symbol domain. Once the frame detection is verified, the frames are processed independently; the function labeled "Viterbi and Reed–Solomon decoders" is replicated as many times as needed to meet the required data rate and latency. This architecture is especially suitable for implementation with parallel processors and for cases where a "pool" of resources is available to cover the needs of many users with diverse needs.

Case 2: Independent frames, restriction on latency. The process of frame detection often requires performing the correlation over multiple frames and bridging "gaps" that eliminate frame markers, therefore introducing a substantial, and sometimes unacceptable, latency. In this case, the architecture can be modified as shown in Fig. 5, utilizing the symbol-domain frame-marker correlator only for tentative frame identification. Decoding of frames starts immediately following this tentative detection. However, results from the other synchronization algorithms are used to verify the synchronization and are fed back to allow correction of the synchronization, if needed.

The extent of input buffering and feedback depends on a trade-off between the latency requirement and the implementation restrictions on data rate and available storage.

Case 3: Nonindependent frames or tight latency requirements. This case occurs either when the frame marker is shorter than the convolutional code constraint length (hence, the encoding is not independent from frame to frame) or when the tight latency requirement dictates that Viterbi decoding must be initiated even prior to

---

[2] Latency is the time delay between the signal arrival at the antenna and the time a fully decoded transport frame is available at the output of the decoder.

[3] E. Greenberg, JPL Interoffice Memorandum 3171-93-20 (internal document), Jet Propulsion Laboratory, Pasadena, California, April 6, 1993.

frame synchronization. The resulting decoder architecture is shown in Fig. 6—the Viterbi decoder relies on its internal measures, e.g., accumulated-metric-growth rate, to achieve node synchronization, but receives feedback from the symbol-domain and bit-domain frame-marker correlators. In this architecture, multiple Viterbi decoders could be employed to expedite the detection of the node synchronization over multiple symbol phase offsets.

IV. Examples

The joint synchronization approach has been applied in two decoder systems developed at JPL. The first decoder, the Maximum-Likelihood Convolutional Decoder Mark III (MCD III) [4,5], is a fully programmable $(K, 1/n)$ Viterbi decoder with $3 \leq K \leq 15$ and $2 \leq n \leq 6$. It is implemented in a fully parallel architecture using 64 identical, custom, very large-scale integrated circuit (VLSI) devices capable of operating at bit rates of up to 1.1 Mbits/sec. The MCD III incorporates all three synchronization algorithms described in Section II, resulting in a flexible synchronization architecture.

A more recent example is the Feedback Concatenated Decoder (FCD) developed specifically for the Galileo mission [6]. This decoder is implemented in software on a four-central-processing-unit (CPU) SUN workstation. It is capable of performing (14,1/4) Viterbi and four-redundancy (255,x) Reed–Solomon decoding as well as the associated redecoding at 160 bits/sec. As an integrated decoder, it is oriented toward symbol-domain frame synchronization, while attempting to minimize latency (case 2 above).

V. Conclusions

We have presented a method of performing joint frame and node synchronization for a concatenated decoder. This approach enables a design that shortens the acquisition time and allows for parallel implementation of the decoder resource-demanding tasks, thus improving the overall decoder efficiency.

References

[1] S. Lin and D. Costello, *Error Control Coding*, Prentice–Hall Series on Computer Applications in Electrical Engineering, Englewood Cliffs, New Jersey: Prentice–Hall, 1983.

[2] Consultative Committee on Space Data Standards, *Recommendation for Space Data Systems Standards: Telementry Channel Coding*, Document 101-0-B-3 (blue book issue 3), Washington, DC: National Aeronautics and Space Administration, May 1992.

[3] J. Statman, "Optimizing the Galileo Space Communication Link," *The Telecommunications and Data Acquisition Progress Report 42-116, vol. October–December 1993*, Jet Propulsion Laboratory, Pasadena, California, pp. 114–120, February 15, 1994.

[4] J. Statman, G. Zimmerman, F. Pollara, and O. Collins, "A Long Constraint Length VLSI Viterbi Decoder for the DSN," *The Telecommunications and Data Acquisition Progress Report 42-95, vol. July–September 1988*, Jet Propulsion Laboratory, Pasadena, California, pp. 186–200, November 15, 1988.

[5] K. M. Cheung, L. Swanson, and S. Arnold, "Node Synchronization Schemes for the Big Viterbi Decoder," *The Telecommunications and Data Acquisition Progress Report 42-108, vol. October–December 1991*, Jet Propulsion Laboratory, Pasadena, California, pp. 134–142, February 15, 1992.

[6] T. Chauvin and K. M. Cheung, "A Parallel Viterbi Decoder for Shared Memory Multiprocessor Architecture," presented at the 3rd SIAM Conference on Linear Algebra in Signals, Systems, and Control, Seattle, Washington, August 1993.
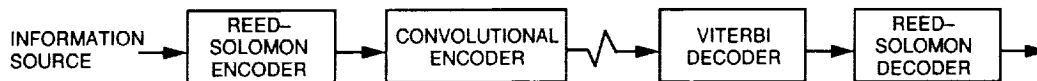
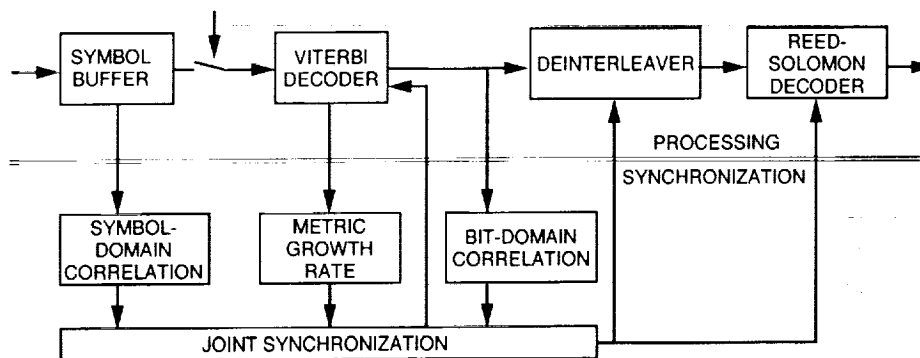Fig. 1. Error-correcting encoder/decoder traditional configuration.

Fig. 2. Joint-synchronization decoder structure.
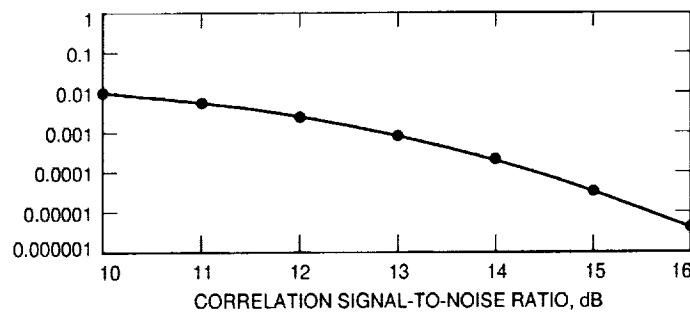

CORRELATION SIGNAL-TO-NOISE RATIO, dB

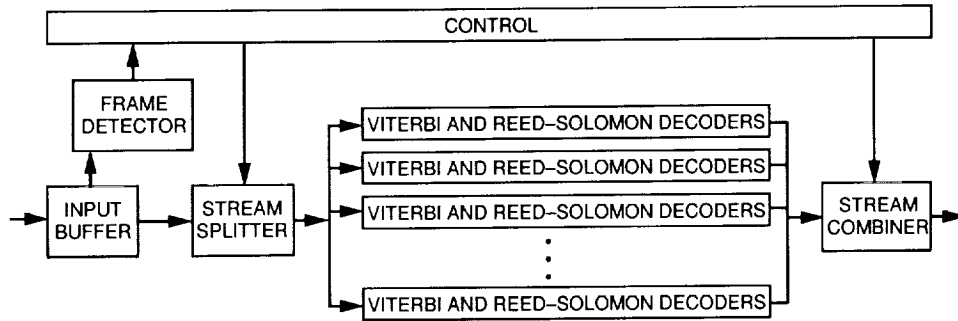Fig. 3. Probability of false detection versus correlation SNR.

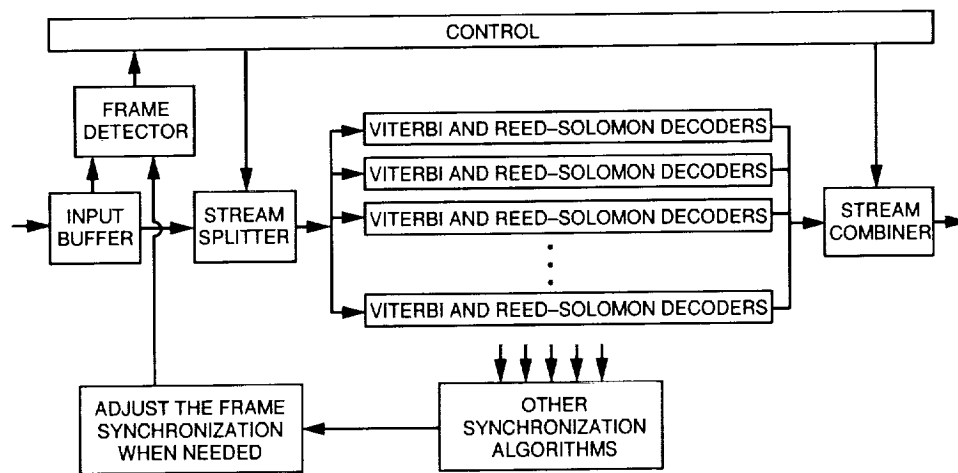Fig. 4. Decoder architecture without synchronization feedback.

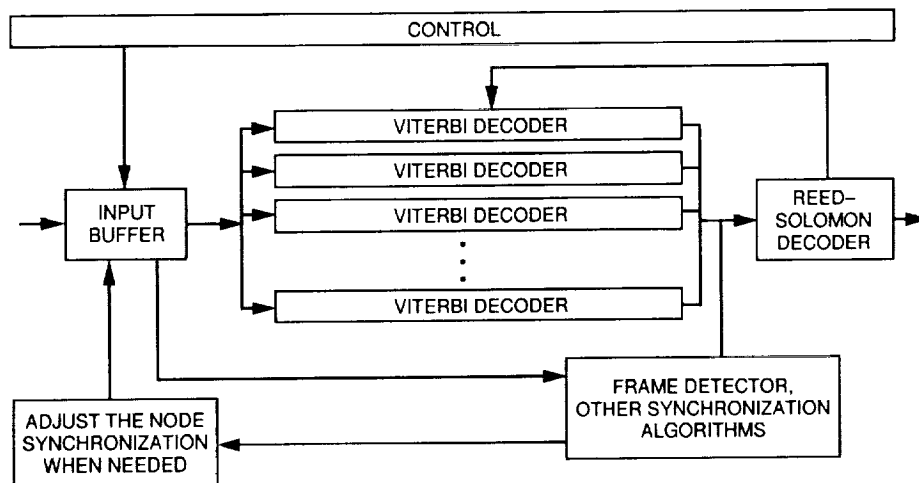Fig. 5. Decoder architecture with feedback for synchronization correction.

Fig. 6. Decoder architecture when the Viterbi decoder is self-synchronized.